**Aurora**

* Overview
  + Aurora is a relational database service that combines the speed and availability of high-end commercial databases with the simplicity and cost-0effectiveness of open source databases
  + Aurora is an AWS proprietary db
  + Fully managed service
  + High performance, low price
  + Scales in 10GB increments
  + Scales up to 32vCPUs and 244GB RAM
  + 2 copies of data are kept in each AZ with a minimum of 3 AZ's(6 copies)
  + Can handle the loss of up to two copies of data without affecting DB write availability and up to three copies without affecting read availability
  + The following diagram depicts how Aurora Fault Tolerance and Replicas work:
    - Diagram

      Description automatically generated with medium confidence
  + You can create read replicas for Amazon Aurora database in up to 5 regions. This capability is available for Aurora with MySQL compatibility
  + Cross-Region Read Replicas
    - Cross-region RR's allow you to improve your disaster recovery posture, scale read operations in regions closer to your application users, and easily migrate from one region to another
    - Cross-region replicas provide fast local reads to your users
    - Each region can have an additional 15 Aurora replicas to further scale local reads
    - You can choose between Global Database, which provides the best replication performance, and traditional binlog-based replication
    - You can also set up your own binlog replication with external MySQL databses
    - The following diagram depicts the Cross-Region Replica topology:
* Diagram

  Description automatically generated
  + Global Database
    - For globally distributed applications you can use Global Database, where a single Aurora databse can span multiple regions to enable fast local reads and quick disaster recovery
    - Global Databse uses storage-based replication to replicate database across multiple AWS Regions, with typical altency of less than 1 second
    - You can use a secondary region as a backup option in case you need to recover quickly from a regional degradation or outage
    - A database in a secondary region can now be promoted to full read/write capabilities in less than 1 minute
    - The following table depicts the Aurora Global Database topology:
* Graphical user interface, application, Word

  Description automatically generated
  + Multi-Master
    - Aurora Multi-Master is a new feature of the Aurora MySQL-compatible edition that adds the ability to scale out write performance across multiple AZ's, allowing applications to direct read/write workloads to multiple instances in a database cluster and operate with higher availability
    - Aurora Multi-Master is designed to achieve HA and ACID transactions across a cluster of database nodes with configurable read after write consistency
  + Architecture
    - An Aurora cluster consists of a set of compute(database) nodes and a shared storage volume
    - The storage volume consists of six storage nodes placed in three AZ's for HA and durability of user data
    - Every database node in the cluster is a write node that can run read and write statements
    - There is no single point of failure in the cluster
    - Applications can use any writer node for their read/write and DDL needs
    - A database chjange made by a write node is written to six storage nodes in three AZ's, providing data durability and resiliency against storage node and AZ failures
    - The write nodes are all functionally equal, and a failure of one write node does not affect the availability of the other writer nodes in the cluster
  + HA
    - Aurora Multi-Master improves upon the HA of the single-master version of Aurora because all the ndoes in the cluster are read/write nodes
    - With single-master Aurora, a failure of the single writer node requires the promotion of a read replica to be the new writer
    - In the case of Aurora Multi-Master, the failure of a writer
  + Serverless
    - Aurora Serverless is an on-demand, auto-scaling configuration for Aurora
    - Available for MySQL-compatible and PostgreSQL-compatible editions
    - The database automatically starts up, shuts down and scales capacity up or down based on the application needs
    - It enables you to run a database in the cloud without managing any database instances. It's a simple, cost-effective option for infrequent, intermittent, or unpredictable workloads
    - You simply create a database endpoint and optionally specify the desired database capacity range and connect applications
    - With Aurora Serverless, you only pay for databse storage and the database capacity and I/O your database consumes while it is active
    - Pay on per-second basis for the database capacity you use when the database is active
    - Can migrate between standard and serverless configurations with a few clicks in the RDS Management Console
    - The table below provides a few example use cases for Aurora Serverless:
  + Table

    Description automatically generated
  + Fault-Tolerant and Self-Healing Storage
    - Each 10GB chunk of your database volume is replicated six ways, across 3 AZ's
    - Aurora storage is fault-tolerant, transparently handling the loss of up to two copies of data without affecting databse write availability and up to three copies without affecting read availability
    - Aurora storage is also self-healing; data blocks and disks are continuously scanned for errors and replaced automatically
  + Aurora Auto-Scaling
    - Aurora Auto Scaling dynamically adjusts the number of Aurora Replicas provisioned for an Aurora DB cluster using single-master replication
    - AS is available for both Aurora MySQL and Aurora PostgreSQL
    - AS enables your Aurora DB cluster to handle sudden increases in connectivity or workload
    - When the connectivity or workload decreases, AS removes the unnecessary Replicas so that you don't pay for unused provisioned DB instances
  + Automatic, Continuous, Incremental Backups and Point-in-Time Restore
    - Aurora's backup capability enables point-in-time recovery for your instance
    - This allows you to restore your database to any second during your retention period, up to the last 5 minutes
    - Your automatic backup retention period can be configured up to 35 days
    - Automated backups are stored in S3, which is designed for 11 9's of durability. Aurora backups are automatic, incremental and continuous and have no impact on database performance
    - When automated backups are turned on for your DB instance, RDS automatically performs a full daily snapshot of your data(during your preffered backup window) and captures transaction logs(as updates to your DB instances are made)
    - Automated backups are enabled by default and data is stored on S3 and is equal to the size of the DB
    - RDS retains backups of a DB instance for a limited, user-specified period of time called the retention period, which default is 7 days, but can be up to 35 days
    - There are two methods to backup and restore RDS DB instances
      * RDS automated backups
      * User initiated manual backups
    - Both options back up the entire DB instance and not just the individual DBs
    - Both options create a storage volume snapshot of the entire DB instance
    - You can make copies of automated backups an dmanual snapshots
    - Automated backups and backup data to multiple AZs to provide for data durability
    - Multi-AZ backups are taken from the standby instance(for MariaDB, MySQL, Oracle and PostgresSQL)
    - The DB instance must be in an Active state for automated backups to happen
    - Only automated backups can be used for point-in-time DB instance recoveery
    - The granularity of point-in-time recovery is 5 minutes
    - RDS creates a daily full storage volume snapshot and also captures transaction logs regularly
    - You can choose the backup window
    - There is no additional charge for backups but you will pay for the storage costs on S3
    - You can disable automated backups by setting the retention period to 0
    - An outage occurs if you change the backup retention period from zero to non-zero value or the other way around
    - **The retnetion period is the period AWS keeps the automated backups before deleting them**
      * By default the retention period is 7 days if ocnfigured form the console for all DB engines except Aurora
      * The default retention period is 1 day if configured from the API or CLI
      * The retention period for Aurora is 1 day regardless of how it is configured
      * You can increase the retention period up to 35 days
    - During the backup window I/o may be suspended
    - Automated backups are deleted when you delete the RDS DB instance
    - Automated backups are only supported for InnoDB storage engine for MySQL(not for myISAM)
    - When you restore a DB instance the default DB parameters and SG's are applied - you must then apply the custom DB parameters and SG's
    - You cannot restore from a DB snapshot into an existing DB instance
    - Following a restore the new DB instance will have a new endpoint
    - The storage type can be changed when restoring a snapshot